EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L4	1	(model$3 near compress$5) and ((triangle or triangular) near mesh) and (B$spline near curve)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 09:41
L5	23	((triangle or triangular) near mesh) and (B$spline near curve)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 09:41
L6	9	((triangle or triangular) near mesh) and (B$spline near curve) and compress$5	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 09:47
L7	9	((triangle or triangular) near mesh) and (B$spline near curve) and compress$5 and (segment$5 or subdivision)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 09:47
L8	9	((triangle or triangular) near mesh) and (B$spline near curve) and compress$5 and (segment$5 or subdivi$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 10:01
L9	1	((triangle or triangular) near mesh) and (B$spline near curve) and compress$5 and (segment$5 or subdivi$5) and watershed	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 09:50
L10	1	((triangle or triangular) near mesh) and (estimat$4 near curve) and (watershed near (segment$5 or subdivi$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 10:02
L11	2	((triangle or triangular) near mesh) and (estimat$4 near curv$5) and (watershed near (segment$5 or subdivi$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 10:03
L12	2	((triangle or triangular) near mesh) and (estimat$4 with curv$5) and (watershed near (segment$5 or subdivi$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 10:03

EAST Search History

L13	2	((triangle or triangular) near mesh) and (estimat$4 with curv$5) and (watershed with (segment$5 or subdivi$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 10:03
L14	2	((triangle or triangular) with mesh) and (estimat$4 with curv$5) and (watershed with (segment$5 or subdivi$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 10:03
L15	18	(estimat$4 with curv$5) and (watershed with (segment$5 or subdivi$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 10:04
L16	2	(estimat$4 with curv$5) same (watershed with (segment$5 or subdivi$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 10:27
L17	2	(estimat$4 with curv$5) same watershed	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 10:22
L20	7	curv$5 same (watershed with (segment$5 or subdivi$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 11:02
L21	1	(weibull adj E$SD adj fields) and segmen$6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 11:03
L22	1	(weibull adj E$SD adj fields)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 11:03
L23	205	(weibull and segmen$6)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 11:04
L24	0	(weibull near fields) and segmen$6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 11:03

EAST Search History

L25	13	(weibull same segmen$6)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 11:21
L26	4	segmen$6 same greedy same refin$5	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 11:21
S9	3124	(generat$3 or build$3 or creat$3) with (model$3 adj data)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 11:36
S10	649	(generat$3 or build$3 or creat$3) near (model$3 adj data)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 09:22
S11	0	((point near co$odinate) with ((3D or three$D or "3"$dimansional or three$dimensional) adj object))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 09:06
S12	1	(co$odinate with ((3D or three$D or "3"$dimansional or three$dimensional) adj object))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 09:08
S13	36	(segment$3 or divid$3 or partition$3 or portion$3) near model$3 near feature$1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 11:24
S14	2	S10 and S13	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 09:17
S15	22	S13 and ((generat$3 or build$3 or creat$3) near model$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 11:16
S16	1	S10 and (watershed adj segment$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 11:35

EAST Search History

S17	97	(watershed adj segment$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 11:35
S18	35	S17 and (generat$3 or build$3 or creat$3) with model$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 12:46
S27	36	(segment$3 or divid$3 or partition$3 or portion$3) near model$3 near feature$1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 11:17
S28	0	S27 and ((generat$3 or build$3 or creat$3) near model$3) and (input with criteria)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 11:17
S29	3130	(generat$3 or build$3 or creat$3) with (model$3 adj data)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 11:19
S30	2	S29 and (input with criteria) and ((search$3 or query or retriev$3) with (object adj feature))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 11:19
S31	2	S29 and ((input with criteria) same ((search$3 or query or retriev$3) with (object adj feature)))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 11:21
S32	2	((input with criteria) same ((search$3 or query or retriev$3) with (object adj feature)))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 11:53
S33	2	((input with criteria) same ((search$3 or query or retriev$3) same (object adj feature)))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 11:22
S34	102	(input same ((search$3 or query or retriev$3) same (object adj feature)))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 12:01

EAST Search History

S35	3	S29 and (input same ((search$3 or query or retriev$3) same (object adj feature)))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 11:22
S36	0	S27 and S34	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 11:24
S37	89	S34 and (segment$3 or divid$3 or partition$3 or portion$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 11:33
S38	11	S34 and ((segment$3 or divid$3 or partition$3 or portion$3) with (object adj feature))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 11:26
S39	78	S37 not S38	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 11:33
S40	2	((input with criteria) same ((search$3 or query or retriev$3) with (object near feature)))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 11:53
S41	2	((input with criteria) same ((search$3 or query or retriev$3) same (object near feature)))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 11:54
S42	2	((input with criteria) same ((search$3 or query or retriev$3) same (feature near object)))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 11:54
S45	99	(input same ((search$3 or query or retriev$3) same (object adj feature))) and (model$3 ajd data)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 12:01
S46	18	(input same ((search$3 or query or retriev$3) same (object adj feature))) and (model$3 adj data)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 12:01

EAST Search History

S60	899	"triangle mesh"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 16:04
S61	74	"triangle mesh" and (B$spline)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 16:05
S62	23	"triangle mesh" and (B$spline near curve$1)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 16:28
S63	1	"triangle mesh" and (B$spline near curve$1) and (subdivi$5 near surface near compress$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 16:38
S64	1	"triangle mesh" and (B$spline near curve$1) and (surface near compress$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 16:07
S65	9	"triangle mesh" and (B$spline near curve$1) and (compress$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 16:07
S66	1	S62 and watershed	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 16:08
S67	3	S62 and hybrid	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 16:10
S68	2	S62 and hybrid and segmen$7	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 16:11
S69	9	"triangle mesh" and (B$spline near curve$1) and compress$5	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 16:28

EAST Search History

S70	1	(hybrid near segmen$7) and (subdivi$5 near surface near compress$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 16:40
S71	1	(hybrid near segmen$7) and (surface near compress$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 16:40
S72	293	(hybrid near segmen$7)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 16:40
S73	1	(hybrid near segmen$7) and (watershed near segmen$7)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/28 16:41
S74	5	(hybrid near segmen$7) and (watershed)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/29 07:50

Color Image Segmentation Solving Hard-Constraints on Graph Partitioning Greedy Algorithms

Full text Publisher Site

Source ICPR archive
Proceedings of the International Conference on Pattern Recognition - Volume 3 table of contents
Page: 3629
Year of Publication: 2000
ISSN:1051-4651

Publisher IEEE Computer Society Washington, DC, USA

Additional Information: abstract

Tools and Actions: Find similar Articles Review this Article
Save this Article to a Binder Display Formats: BibTex EndNote ACM Ref

ABSTRACT

In this paper, a graph partitioning greedy algorithm is presented. This algorithm avoids the hard-constraints of others similar approaches such as the impossibility for some regions to grow after certain step of the algorithm and the uniqueness of the solution. Nevertheless, it allows attaining global results by local approximations using a generalized concept of not over-segmentation, which includes an energy function, and eliminating the not sub-segmentation criterion using a probabilistic criterion similar to that of annealing. The high-variability region problems such as borders are also eliminated identifying them and distributing their pixels among the other neighbor regions. Thus, it is possible to keep the time complexity of usual graph partitioning greedy algorithm and avoiding its high-variability region problems, obtaining better results.

Web Images Video News Maps **more »**

Search

Advanced Scholar Search
Scholar Preferences
Scholar Help

Scholar

Results **1** - **6** of **6**. (**0.08** seconds)

Sketch-based images database retrieval
S Matusiak, M Daoudi, T Blu, O Avaro - Springer
Page 1. S. Jajodia, MT Özsu, and A. Dogac (Eds.): MIS'98, LNCS 1508, pp. 185-191, 1998. + Springer-Verlag Berlin Heidelberg 1998 ...
Cited by 20 - Related Articles - Web Search - BL Direct

Sketch-based images database retrieval
S MATUSIAK, M DAOUDI, T BLU, O AVARO - Lecture notes in computer science - cat.inist.fr
Sketch-based images database retrieval. S MATUSIAK, M DAOUDI, T BLU, O AVARO Lecture notes in computer science, 185-191, Springer. ...
Web Search

Sketch-Based Images Database Retrieval
S Matusiak, M Daoudi, T Blu, O Avaro - Proceedings of the 4th International Workshop on Advances in ..., 1998 - portal.acm.org
Google, Inc. Subscribe (Full Service), Register (Limited Service, Free), Login. Search: The ACM Digital Library The Guide. Feedback ...
Web Search

[PS] Sketch-Based Images Database Retrieval
S Matusiak, M Daoudi, T Blu, O Avaro - wotan.liu.edu
Page 1. _____ I NTERNATIONAL W ORKSHOP ON "M ULTIMEDIA ...
View as HTML - Web Search

Sketch-Based Images Database Retrieval
S Matusiak, M Daoudi, T Blu, O Avaro - Proceedings of the Fourth International Workshop on - bigwww.epfl.ch
EPFL. Publications. English only, BIG > Publications > Database Retrieval. ...
Cached - Web Search

[PS] Sketch-Based Images Database Retrieval
S Matusiak, M Daoudi, T Blu, O Avaro - tele.ucl.ac.be
Page 1. _____ I NTERNATIONAL W ORKSHOP ON "M ULTIMEDIA ...
View as HTML - Web Search

Search

Google Home - About Google - About Google Scholar

IEEE TRANSACTIONS ON
VISUALIZATION AND COMPUTER GRAPHICS

October-December 1999 (Vol. 5, No. 4) pp. 308-321

Partitioning 3D Surface Meshes Using Watershed Segmentation

Alan P. Mangan
Ross T. Whitaker

DOI Bookmark:

http://doi.ieeecomputersociety.org/10.1109/2945.817348

Abstract

This paper describes a method for partitioning 3D surface meshes into useful segments. The proposed method generalizes morphological watersheds, an image segmentation technique, to 3D surfaces. This surface segmentation uses the total curvature of the surface as an indication of region boundaries. The surface is segmented into patches, where each patch has a relatively consistent curvature throughout, and is bounded by areas of higher, or drastically different, curvature. This algorithm has applications for a variety of important problems in visualization and geometrical modeling including 3D feature extraction, mesh reduction, texture mapping 3D surfaces, and computer aided design.

References

About References | Back to Top

[1] J.P. Serra, *Image Analysis and Mathematical Morphology*.

London: Academic Press, 1982.

[2] H. Hoppe, "Progressive Meshes," *Proc. SIGGRAPH '96*, pp. 99-108, 1996.

[3] A. Varshney, P. Agarwal, and F.P.B. Jr, "Automatic Generation of Multiresolution Hierarchies for Polygonal Models," *Proc. First Workshop Simulation and Interaction in Virtual Environments*, pp. 25-28, 1995.

[4] A.W.F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin, "MAPS: Multiresolution Adaptive Parameterization of Surfaces," *Computer Graphics (SIGGRAPH '98 Proc.)*, M. Cohen, ed., vol. 32, pp. 95-104, July 1998.

[5] C.L. Bajaj and T. Dey, "Convex Decomposition of Polyhedra and Robustness," *SIAM J. Computing*, vol. 21, pp. 339-364, 1992.

[6] B. Chazelle and L. Palios, "Triangulating a Nonconvex Polytope," *Discrete and Computational Geometry*, vol. 5, pp. 505-526, 1990.

[7] B. Chazelle, D.P. Dobkin, N. Shouraboura, and A. Tal, "Strategies for Polyhedral Surface Decomposition: An Experimental Study," *Proc. 11th Ann. ACM Symp. Computational Geometry*, pp. 297-305, June 1995.

[8] J. Koënderink and A. van Doorn, "The Structure of Two-Dimensional Scalar Fields with Applications to Vision," *Biological Cybernetics*, vol. 33, pp. 151-158, 1979.

[9] D. Eberly, *Ridges in Image and Data Analysis*. Dordrecht: Kluwer Academic, 1996.

[10] R.B. Fisher, A.W. Fitzgibbon, and D. Eggert, "Extracting Surface Patches from Complete Range Descriptions," *Proc. Int'l Conf. Recent Advances in 3-D Digital Imaging and Modeling*, pp. 148-154, Ottawa,

Canada, May 1997.

[11] D. Faugeras and M. Hebert, "A 3-D Recognition and Positioning Algorithm Using Geometric Matching between Primitive Surfaces," *Proc. Eighth Int'l Joint Conf. Artificial Intelligence,* pp. 996-1,002, 1983.

[12] P.J. Besl, *Surfaces in Range Image Understanding.* New York: Springer-Verlag 1988.

[13] E. Trucco and R.B. Fisher, "Experiments in Curvature-Based Segmentation of Range Data," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 2, pp. 177-181, Feb. 1995.

[14] M. Baccar, "Surface Characterization Using a Gaussian Weighted Least Squares Technique towards Segmentation of Range Images," master's thesis, Univ. Tennessee, K noxville, May 1994.

[15] A.C.F. Colchester, "Network Representation of 2D and 3D Images," *3D Imaging in Medicine,* K.H. Höhne, H. Fuchs, and S. Pizer, eds., pp. 45-62, Springer-Verlag, 1990.

[16] L.D. Griffin, A.C.F. Colchester, and G.P. Robinson, "Scale and Segmentation of Gray-Level Images Using Maximum Gradient Paths," *Image and Vision Computing,* vol. 10, pp. 389-402, 1992.

[17] L.R. Nackerman, "Two-Dimensional Critical Point Configuration Graphs," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 6, no. 4, pp. 442-450, 1984.

[18] J. Koënderink and A. van Doorn, "Local Features of Smooth Shapes: Ridges and Courses," *SPIE Proc. Geometric Methods in Computer Vision II,* vol. 2031, pp. 2-13, 1993.

[19] W.E. Lorensen and H.E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Computer Graphics (SIGGRAPH '87 Proc.),* vol. 21, pp. 163-169, 1987.

[20] J. Koënderink, *Solid Shape*. Cambridge, Mass.: MIT Press, 1991.

[21] J. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Science*. New York: Cambridge Univ. Press, 1996.

[22] R.T. Whitaker, "A Level-Set Approach to 3D Reconstruction From Range Data," *Int'l J. Computer Vision*, vol. 29, pp. 203-231, Oct. 1998.

[23] C. Gourley, "Pattern Vector Based Reduction of Large Multimodal Data Sets for Fixed Rate Interactivity during Visualization of Mutliresolution Models," PhD thesis, Univ. of Tennessee, K noxville, 1998.

[24] C. Gourley, C. Dumont, and M.A. Abidi, "Fixed-Rate Interactivity for Visualization of Photo-Realistic Multiresolution Models," *Am. Nuclear Soc.: Eighth Topical Meeting on Robotics and Remote Systems*, Apr. 1999.

Additional Information

Back to Top

Index Terms- Surfaces, surface segmentation, watershed algorithm, curvature-based methods.

Citation: Alan P. Mangan, Ross T. Whitaker, "Partitioning 3D Surface Meshes Using Watershed Segmentation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 05, no. 4, pp. 308-321, Oct-Dec, 1999.

Computer-Aided Design
Volume 35, Issue 9 , August 2003, Pages 783-789

doi:10.1016/S0010-4485(02)00101-X Cite or Link Using DOI

A hybrid approach to feature segmentation of triangle meshes

Anshuman Razdan[a] and **MyungSoo Bae**[b]

[a] PRISM, Arizona State University, MC 5906, Tempe, AZ 85287-5906, USA
[b] PRISM and Computer Science and Engineering, Arizona State University, Tempe, AZ 85287-5906, USA

Received 28 November 2001; revised 16 May 2002; accepted 18 May 2002. ; Available online 11 September 2002.

Abstract

Segmentation of a polygonal mesh is a method of breaking the mesh down into 'meaningful' connected subsets of meshes called regions or features. Several methods have been proposed in the past and they are either vertex based or edge based. The vertex method used here is based on the *watershed segmentation scheme* which appears prominently in the image segmentation literature and was later applied to the 3D segmentation problem [9 and 10]. Its main drawback is that it is a vertex based method and no hard boundaries (edges) are created for the features or regions. Edge based methods rely on the dihedral angle between polygon faces to determine if the common edge should be classified as a *Feature Edge*. However, this method results in many disconnected edges and thereby incomplete feature loops.

We propose a hybrid method which takes advantage of both methods mentioned earlier and create regions with complete feature loops. Satisfactory results have been achieved for both CAD parts as well as other laser scanned objects such as bones and ceramic vessels.

Author Keywords: Three-dimensional segmentation; Shape recognition; Feature extraction;

Watershed segmentation; Curvature estimation; Triangle mesh

Article Outline

1. Introduction

The domain of the problem, a triangle mesh (denoted by M^1), consists of a set of *n* points (vertices $\mathbf{v}_i \in \mathbb{E}^3$; $0 \leqslant i < n$) and a set of planar convex polygons (faces) made up of these vertices. These are two-dimensional manifolds representing the surface of a three-dimensional object. Triangle meshes are currently a popular surface modeling primitive used extensively to represent real world and synthetic surfaces in computer graphics.

Segmentation means breaking down an existing structure into meaningful, connected subcomponents. In the context of meshes, the sub-components of the structure being broken down are sets of vertices called *regions* which share some 'commonality'. Mesh segmentation has had numerous applications in the past. Moreover, several existing applications benefit from segmentation. Good examples are feature extraction, decimation, adaptive subdivision [1], and surface fitting for reverse engineering. The motivation for this research was provided by two application domains: optimized triangulated meshes in CAD (Reverse Engineering, Manufacturing, etc.) and over sampled meshes from laser scanned data such as ceramic vessels and bones. Since curvature is not scale invariant, we scale all input files to a unit-bounding box.

2. Related work

We first present two approaches that have been used for segmentation of polygonal meshes and then present our method.

2.1. Vertex based method

A value λ_i could be associated with each vertex $\mathbf{v}_i$ in the data set which somehow

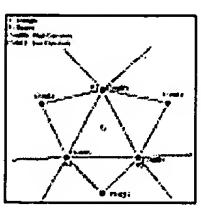
encapsulates the characteristics of the locality of the vertex. The definition of segmentation is the one in which regions consist of connected vertices which have the same λ (within a tolerance). Curvature is selected as the mathematical basis for region separation, i.e. the scalar value λ. Curvature estimation from 3D meshes is dealt with in Hamann [5], Calladine [3] and Kobbelt [8], who extract curvature from a locally fitted quadratic polynomial approximant; and Besl [2], Hoschek [7], and Mangan and Whitaker [9], who describe various discrete curvature approximation schemes. Vertices having the same curvature value would be grouped into regions, separated by vertices with high curvature (which serve as region boundaries). An improved curvature estimation scheme is presented in [10].

The segmentation scheme used is derived from the *Watershed* algorithm for 3D meshes described in [9]. However, there has been considerable research work relevant to this problem based on various other techniques [13, 4, 11, 2 and 6].

Mangan and Whitaker [9] generalize the watershed method to arbitrary meshes by using either the discrete Gaussian curvature or the norm of covariance of adjacent triangle normals at each mesh vertex as the *height field*, analogous to the pixel intensity on an image grid which drives the 2D watershed segmentation algorithm. From our experience, absolute curvature provides the best results for Watershed segmentation [10].

2.1.1. The no hard boundary problem

A major drawback of the vertex-based method is that no hard boundaries are created for the features or regions. Each vertex of an object has its own region information. Therefore, triangles on boundaries have multi-region information. The three vertices of a triangle can be part of three different regions, whereas the triangle itself would be a 'gray' area, i.e. it would not belong to any one region (Fig. 1). Fig. 2(a) shows the boundary triangles in white. This means the regions will not have hard boundaries or edges. This is a known artifact of vertex based Watershed segmentation and is acknowledged in [9]. We call this the *no hard boundary* problem. To solve this problem, we create new triangles by adding midpoints to the edges which have different region labels. Each new vertex contains multiple labels (Fig. 3). This is an extension of the original algorithm [9]. For the selection of the label(s) of a vertex which has multiple labels, the common label of the vertices of the triangle is selected, Fig. 2(b). In Fig. 3(a), there are two possible diagonal splits to make triangles. We select the diagonal which satisfies $\max(\min_{1\leqslant i\leqslant 4} a_i, \min_{1\leqslant i\leqslant 4} b_i)$ where a_i and b_i are interior angles formed by the diagonals, i.e. select the diagonal that results in the best aspect ratio.



(5K)

Fig. 1. Shows an example where a triangle T_i will be the gray area, i.e. it will not be part of any region.

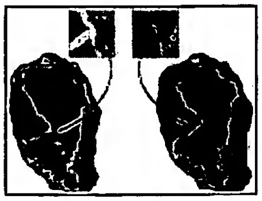
 (19K)

Fig. 2. Segmentation using Watershed method: (a) with no hard boundaries and (b) with boundary solution.

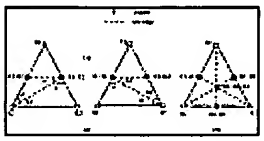
 (7K)

Fig. 3. Creation of triangles on boundaries of an original mesh: (a) for a triangle shared by two regions and (b) for a triangle shared by three regions.

Triangle meshes representing mechanical (CAD) objects are frequently sparse in some areas, with just enough vertices to define each area. This is usually a result of the optimal triangulation from a CAD program or decimation process. In this case, the Watershed method may not segment the object properly or may even lose important regions on the objects. In Fig. 4, the main regions of the object are treated as boundaries because there are not enough vertices in the regions. The boundary solution mentioned earlier will not solve this problem because the method does not create new regions from the boundary regions and the boundary regions will be lost. Moreover, some regions of this object are not segmented properly. This problem is caused by the vertices on feature edges with similar curvatures and those vertices may be treated as the same region. We solve this problem with our proposed approach in Section 3.

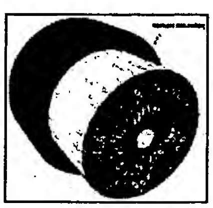
 (8K)

Fig. 4. The mesh segmented using the Watershed method.

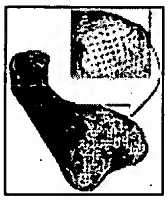
2.2. Feature segmentation based on dihedral angle

This method uses an edge-based method for defining feature boundaries. A Feature Edge is defined as follows. *Feature Edge*: An edge shared by two planes whose normal vectors make an angle greater than a certain threshold. The edges obtained are integrated into curves, and

these curves are classified as jump boundaries, folds (roof edges) and ridge lines [12]. Jump boundaries and folds are used to segment the mesh into several regions. The boundary lines are also treated as Feature Edges.

2.2.1. Shortcomings of the method

The main disadvantage of the Feature Edge-based method is that this results in many disconnected Feature Edges and thereby incomplete Feature Loops. Fig. 5 shows this problem. Feature Edges are shown in brown color.

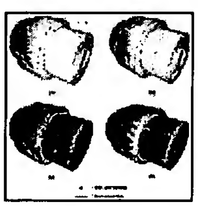


(10K)

Fig. 5. The feature vertices and edges of the mesh.

3. Hybrid approach

Our proposed method takes advantages of both methods mentioned in the previous sections by using them in conjunction and hence is called the hybrid method. This creates regions with complete Feature Loops. Fig. 7 shows the steps of the Hybrid method. It is described as follows.



(12K)

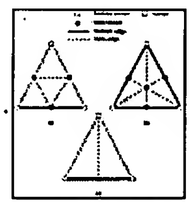
Fig. 7. Various steps of the Hybrid method: (a) original mesh with feature vertices and edges, (b) modified mesh with feature edges and addition of vertices, (c) segmentation using Watershed method, and (d) mesh after deleting additional vertices and triangles.

As explained before, optimally triangulated meshes pose problems for the Watershed segmentation method. To overcome this we identify all the feature edges in the mesh using a threshold (angle). First, we define it as Feature Vertex.

Feature Vertex. Vertices that make up a Feature Edge are defined as Feature Vertices. The reverse is not necessarily true. It is important to note that if both vertices of an edge are Feature Vertices, it does not automatically qualify the edge as a Feature Edge.

Step 1. *Identification of Feature Vertex.* Identify all Feature Vertices based on a threshold angle (dihedral angle) as explained in Section 2.2.

Step 2. *Addition of vertices.* The next step in the process is to add new vertices. Vertices are added to edges of all triangles which have the property that all three vertices are Feature Vertices. See Fig. 6(a). The new vertex is added at the mid point of each edge. We then connect them as shown to create four new triangles. If the triangle has three Feature Edges, then the center point of the triangle is added and six new triangles are created as shown in Fig. 6(b). An addition of vertices requires fixing of the topology as shown in Fig. 6(c). The triangle shown is a neighbor of the triangle to which we added the vertices. This can lead to a hanging vertex problem. To fix this, we connect the new vertex with the vertex on the opposite edge to create two new triangles. As a result of the above, we have two kinds of new vertices; those that lie on the Feature Edges (labeled FV_{high}) and those that do not (labeled as FV_{low}). The reason for labeling is explained below.



(4K)

Fig. 6. Creation of triangles on feature edges of an original mesh.

Step 3. *Watershed segmentation.* Next, we apply Watershed segmentation to our modified mesh. We use absolute curvature for our examples. Other curvature estimation methods may be used and their relative merits are explained in [10]. Feature Vertices FV_{high} are assigned the label of maximum curvature. Since they lie on a Feature Edge, assigning them high curvature ensures that a Feature Edge will be preserved as a hard edge. The rest of the vertices in the mesh follow the same procedure as described in the Watershed algorithm for computing curvatures at the vertices. The Feature Vertices contain their own region labels as well as labels of the neighboring vertices. The addition of vertices has an impact on the Descent and Region Merge operations of the Watershed process [9]. This is done to solve the no hard boundary problem which has been described and discussed earlier in this paper.

Step 4. *Removing vertices added in step 2.* To restore the mesh to its original form we must remove the vertices added to the mesh in step 2 earlier. This process restores the topology of the mesh also.

Step 5. *Collating triangles into regions.* The goal of this step is to assign triangles and not vertices to different regions. This is achieved as follows:

Case 1. *All vertices have the same label.* This is simplest of the cases. The triangle is assigned the region label of its vertices.

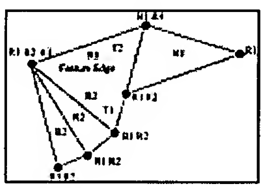
Case 2. *One vertex has a single label.* This is the case when one vertex has a unique label but the other two vertices have multiple labels. The triangle is assigned the region of the vertex

with single label.

Case 3. *Multiple labels but only one common label.* The three vertices of the triangle have multiple labels each, however, there is only one label that is common. The triangle is assigned the region label that is common to the vertices in the triangle.

Case 4. *All edges are Feature Edges.* The triangle qualifies as a region by itself and gets assigned a unique region identifier.

Case 5. *Multiple labels and multiple common labels.* It is possible that each vertex of a triangle has multiple labels and there is more than one common label. To explain this we will use the example in Fig. 8. Triangle T_1 has vertices with region labels R_1 and R_2. One of the vertices also has the label R_3. First, we check if a neighboring triangle shares a Feature edge with T_1. In this case, we find *T2* shares common Feature edge with T_1. We then compare the common vertex labels of $T_1(R_1\ R_2)$ with common label(s) of $T_2(R_1)$. The label that does not belong to the set of common vertex label(s) of the neighboring triangle is assigned to the targeted triangle (R_2).



(3K)

Fig. 8. Assigning regions for case 5, multiple labels with multiple common labels.

If the targeted triangle has more than one feature edge, then the earlier process is repeated for each neighboring triangle that shares a feature edge.[2] Each such neighbor contributes one or more labels. Then T_1 is assigned the label that is common between the contributed labels. The targeted triangle may have no feature edge. It means that the triangle is in the same region as the regions of the neighboring triangles. So, the region label of any neighboring triangle is selected as the label of the targeted triangle.

4. Results and conclusions

The hybrid method relies on the Watershed method and additionally uses advantages of the dihedral angle method. The hybrid method does segmentation of smooth objects as well as mechanical objects. It also solves the no hard boundary problem. Fig. 9 shows two mechanical parts. Fig. 9(a) is a gear and Fig. 9(b) is a housing. Fig. 10(a) shows a turbine. Notice that each blade has been segmented into a different region. Fig. 10(b) is a wheel (rim and tire). It is segmented into the following regions, i.e. the tire, rim, bolt holes and a hole for mounting the wheel. These are correctly segmented. All examples are from public domain data sets. Fig. 11 shows the bone data (from Fig. 5) segmented using the hybrid method.

 (6K)

Fig. 9. Segmentations of mechanical parts using the Hybrid method. Threshold for Feature edge: (a) 40° and (b) 30°. Threshold for curvature: both (a) and (b) 0.1. Both parts were scaled to a unit cube.

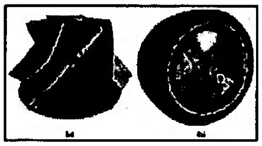
 (6K)

Fig. 10. Segmentation of (a) a turbine and (b) a wheel using the Hybrid method. The threshold for Feature edge: (a) 25° and (b) 30°. Threshold for curvature: both (a) and (b) 0.1. Both parts were scaled to a unit cube.

 (7K)

Fig. 11. Segmentation of the bone using the Hybrid method. Threshold for feature edge: 30°, threshold for curvature: 0.12.

We have pointed out the shortcomings of the most popular segmentation approaches for triangle meshes and devised a hybrid method that overcomes these shortcomings. Future work would involve automatic selection of threshold values both for watershed and dihedral angles.

Acknowledgements

This work was supported in part by the National Science Foundation (grant IIS-9980166). For more information on the 3D Knowledge project, visit http://3dk.asu.edu. The authors would like to thank various members of the Partnership for Research In Stereo Modeling (PRISM) team at Arizona State University for their support.

References

1. Amresh A. Adaptive subdivision schemes for triangular meshes. MS Thesis, Arizona State

University; December 2000.

2. P.J. Besl. *Surfaces in range image understanding*, Springer, Berlin (1988).

3. C. Calladine, Gaussian curvature and shell structures. In: J. Gregory, Editor, *The mathematics of surfaces*, Clarendon Press, Oxford (1986), pp. 179–196.

4. J.-T. Fan, G. Medioni and R. Nevatia, Segmented descriptions of 3D surfaces. *IEEE J Robot Automat* **3** 6 (1987), pp. 527–538.

5. B. Hamann, Curvature approximation for triangulated surfaces. *Comput Suppl* **8** (1993), pp. 139–153.

6. R. Hoffman and A.K. Jain, Segmentation and classification of range images. *IEEE Trans Pattern Anal Mach Intell* **9** 5 (1987), pp. 608–620.

7. Hoschek J, Lasser D. Grundlagen der Geometrischen Datenverarbeitung Teubner BG, Stuttgart; 1989. English translation: Fundamentals of computer aided geometric design, Peters AK; 1993.

8. L. Kobbelt, Discrete fairing and variational subdivision for free-form surface design. *Visual Comput* **16** 3/4 (2000), pp. 142–158. **Full Text** via CrossRef

9. A. Mangan and R. Whitaker, Partitioning 3D surface meshes using watershed segmentation. *IEEE Trans Visualizat Comput Graph* **5** 4 (1999).

10. Pulla S, Razdan A, Farin G. Improved curvature estimation for watershed segmentation of 3-dimensional meshes. Submitted for publication.

11. N.S. Sapidis and P.J. Besl, Direct construction of polynomial surfaces from dense range images through region growing. *ACM Trans Graph* **14** 2 (1995), pp. 171–200. **Full Text** via CrossRef

12. Srikantiah R. Multi-scale surface segmentation and description. MS Thesis, Ohio State University; 2000.

13. M. Yang and E. Lee, Segmentation of measured point data using a parametric quadric surface approximation. *Comput Aided Des* **31** (1999), pp. 449–457. SummaryPlus | **Full Text + Links** | PDF (1541 K)

Corresponding author. Tel.: +1-480-965-5368; fax: +1-480-965-8692

---

[1] A mesh consisting of convex polygons can always be converted to a triangle mesh.

[2] Note that if all three edges are feature edges, then it is dealt under Case 4.

Vitae

Razdan: **Anshuman Razdan** is the Director of PRISM: Partnership for Research In Stereo Modeling at Arizona State University. His research interests include Computer-aided geometric design (CAGD) and Computer Graphics; NURB curves and surfaces Approximation, feature segmentation and use of high bandwidth networking for scientific visualization. He is a PI on several NSF grants including a recent $2.1M KDI grant on 3D Knowledge: Acquisition, Representation and Analysis (3DK). He has a BS and MS in Mechanical Engineering and PhD in Computer Science.

Bae: **MyungSoo Bae** is currently enrolled in the PhD program in the Department of Computer Science and Engineering at Arizona State University. His research interests lie broadly in the area of CAGD. These include surface matching and metrics for meshes, feature segmentation and surface approximation using NURBS. He holds a MS in Computer Science from Arizona State University.

Computer-Aided Design
Volume 35, Issue 9 , August 2003, Pages 783-789